



LAWRENCE  
LIVERMORE  
NATIONAL  
LABORATORY

# Notes from the Unit Commitment Linear Programming Project

B. L. Rountree

August 10, 2012

## **Disclaimer**

---

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

# Notes from the Unit Commitment Linear Programming Project

Barry Rountree

April 9, 2012

## Abstract

The Unit Commitment Problem (UCP) is a computational bottleneck in electrical grid optimization. The problem bears a close resemblance to a similar problem in High Performance Computing: the Dynamic Voltage/Frequency Scheduling (DVFS) problem. In my earlier work I successfully transformed the DVFS into a closely related problem amenable to a linear programming solution and was thus able to schedule problem instances several orders of magnitude larger than any that had been attempted before. This technical report details my unsuccessful efforts to derive a similar transformation for the UCP. The problem proved impossible not because generators were either on or off, but because the state of an earlier timeslice constrains the states of latter timeslices in ways that cannot be sufficiently well approximated for useful solutions.

## 1 Introduction

The general *Unit Commitment Problem* (hence UCP) requires scheduling resources to meet a specified demand over time. Resources are restricted to being either on or off. This constraint creates a natural fit for integer programming formulations. An MILP formulation for UCP will partition the problem into timeslices and schedule sufficient resources per timeslice while minimizing cost. Despite extraordinary advances in ILP and MILP solvers this kind of formulation guarantees all but the smallest problems will be intractable.

Dozens of approaches have been formulated since the unit commitment problem was first formalized [13]. There are two major reviews of this literature [25,37]. Table 1 illustrates how nearly every novel optimization technique developed over the past fifty years has eventually been applied to UCP. This is a testament to both the importance of the problem and the perception that existing techniques leave significant room for improvement.

In my previous work [32] I was faced with a similar problem: creating an optimal schedule for binary resources, in this case the selection of discrete CPU clock frequencies in order to minimize energy consumption during execution of parallel scientific applications. Earlier work had treated this as a classic

Table 1: Parameters

Genetic Programming	[1, 4, 5, 9, 14, 17, 24, 28, 31, 33, 38, 40, 44, 46, 49]
Lagrangian Relaxation	[2, 3, 7, 12, 18, 21, 27, 34, 46, 47, 50, 51]
Dynamic Programming	[8, 11, 13, 22, 24, 26, 42]
Simulated Annealing	[15, 17, 29, 41, 52]
Neural Networks	[9, 19, 23, 36, 48]
Fuzzy Algorithms	[24, 35, 43]
Tabu Algorithms	[16, 17, 30]
Expert Systems	[20, 24, 40]
Particle Swarm	[6, 45]
Unit Period	[39]
Evolutionary Algorithms	[10]

integer programming problem: only a single CPU clock frequency could be used at a time and optimization required selecting the best frequency from the handful of candidate frequencies supported by the processor. The overhead due to integer programming precluded using this work for fine-grained MPI scheduling necessary for massively parallel programs.

I successfully solved this problem by assuming that every CPU clock frequency would be used between MPI communication calls and set up the solver to determine when execution should begin in each frequency. In this case, time became my continuous variable. The optimal solutions never required more than two frequencies to be scheduled for a positive length of time for any task, and allowing the two frequencies to run one after the other allowed the solution to approximate what would have been the idea CPU clock frequency in terms of both execution time and energy consumption.

The transformation was not perfect. Some amount of time and energy had to be expended in order to change from one frequency to another. If two adjacent computation tasks were to use the same frequency then this cost would not accrue. As this was an either/or variable there was no tractable way to represent it using a linear programming formulation, but the error this introduced was sufficient small so as to not materially affect our solutions.

In attempting a similar formulation for the UCP I hoped that a linear programming approximation could be made to have similarly small error. This was not to be the case. As I explain more fully below, the state of a generator at a particular timeslice may be fully determined by the state of that generator at an earlier timeslice. In contrast, the DVFS problem had been stateless: while I may have underestimated the cost due to changing CPU clock frequency the frequency chosen by a particular task was otherwise wholly independent of the frequency chosen by any task that came before it.

I was able to come up with several tricks compensate for the inability of the linear programming to cope with binary state, but none of them worked sufficiently well — even on toy problems — to allow the solution to have any real-world application. This was the ultimate reason why the project consumed

so much effort: while it was obvious that no linear program could be constructed that would give exactly the same answer as an integer program, the space of nearly-equivalent linear programs was quite large. In the remainder of this technical report I provide several highlights from that space.

## 2 Overview

My most useful formulation of the UCP trades intuitiveness for simplicity. The problem defines *generators*, each of which has a vector of *jobs*. A job represents a change in the state of the generators. Jobs are defined by power (in watts), cost and start times (all jobs run to the end of the time horizon  $T$ ). Most generators will contain jobs with alternating positive and negative watt values. The first job of a particular generator might have an associated power of 10MW while the second job has an associated power of  $-10$ MW. The start of the second job indicates the generator has halted, thus the sum of the first two jobs on this generator anytime after the second job has started is zero.

Load is represented by a single dedicated “generator” generating “negative watts”. The load curve is represented by a stepwise function where each step is a job. As this function is an input the start times for jobs on this generator are known ahead of time. For example, if the initial load is 5MW and it increases an hour later to 7MW, the first job on the generator will have a start time of 0 and “generate”  $-5$ MW and the second job will have a start time of 60 and “generate”  $-2$ MW, thus leading to a total load from this generator of  $-7$ MW at time  $t > 60$ . Decreasing load is represented by the start of a job with positive watts.

The number of jobs per generator (and their characteristics, except for their start times) is an input to the problem. An optimal solution might require a particular generator not to run at all, or to turn on and off a larger number of times than can be represented given the available number of jobs on that generator. We address this problem by optimizing for a time window in the middle of the overall time horizon. Sufficient energy constraints (detailed below) only apply within this window, and any job execution outside this window does not accrue any cost. This will allow only the jobs needed for optimal execution to be scheduled within the window and any unnecessary jobs will be scheduled to run outside of the window. This does require some care in sizing the window and time horizon relative to any additional job constraints. In addition, feasible solutions that require all jobs on a particular generator to be started within the window may benefit from additional jobs on that generator; a feasible solution is not proven to be optimal until there are excess jobs on all generators.

Given this system we would like to be able to sum up the contributions of all jobs whenever a job starts and verify net system power is nonnegative. Doing this based on power would require representing jobs as either on or off — something not generally considered possible in an LP. Instead, we ask how long every job has been running at a particular point in time, multiply that time by the power associated with the job to arrive at the net system energy at that

point in time — a value that should also be nonnegative. While this constraint is necessary it is also insufficient: a early spike in energy production will be averaged across the whole time period, leading to a feasible but nonsensical solution. Instead, we need to constraint the solution so that net system energy between any two adjacent points in time is nonnegative.

These two operations — calculating how long a job has been running and finding net energy between two points in time — is complicated by the fact that jobs are not ordered across generators. The operation we would like to have is *max*: for two points in time  $a$  and  $b$  define  $\delta(a, b)$  as  $\max(a - b, 0)$ . As *max* is not a linear function, we instead put a floor on the desired value:

$$\begin{aligned}\delta(a, b) &\geq a - b \\ \delta(a, b) &\geq 0\end{aligned}$$

Given this, the difficulty then becomes putting an effective ceiling on  $\delta(a, b)$ .

### 3 Basic Formulation

Let there be a set of generators  $G$  and let generator  $g \in G$  consist of  $N_g$  jobs. Next, for  $g \in G$  let  $g_i$  be the  $i$ th job on generator  $g$  (with  $1 \leq i \leq N_g$ ). For  $g \in G$ ;  $i \in 1 \dots N_g - 1$ ;  $n = N_g$  :

$$g_1 = 0 \tag{1}$$

$$g_n = T \tag{2}$$

$$g_i < g_{i+1} \tag{3}$$

These respectively constrain the first job on every generator to start at time zero, the final job on every generator to start at the time horizon  $T$  (the maximum time in the system), and order job start times within a particular generator.

We then define the delta function. Intuitively,  $\delta(x, y)$  is how long job  $y$  executed up to the start of job  $x$ . Outside of a linear programming environment this could be expressed as  $\max(x - y, 0)$ , but within an LP this must be approximated using constraints. Within a single generator this is trivial. For  $g \in G$ ;  $i, j \in 1 \dots N_g$ ,

$$\delta(g_i, g_j) = \begin{cases} g_i - g_j & \text{if } i > j \\ 0 & \text{if } i \leq j \end{cases} \tag{4}$$

Across generators we can only fix the floor, as jobs are not ordered across generators. For  $g, h \in G$ ;  $j \in 1 \dots N_g$ ;  $k \in 1 \dots N_h$  :

$$\delta(g_j, h_k) \geq 0 \tag{5}$$

$$\delta(g_j, h_k) \geq g_j - h_k \tag{6}$$

This establishes a minimum time the  $k$ th job on generator  $h$  ran between time  $t = 0$  and the start of the  $j$ th job on generator  $g$ . However, the sufficient

energy constraint requires knowing how long a job ran between two arbitrary points. For  $f, g, h \in G; i \in 1 \dots N_f; j \in 1 \dots N_g; k \in 1 \dots N_h$  :

$$\gamma(f_i, g_j, h_k) \geq 0 \quad (7)$$

$$\gamma(f_i, g_j, h_k) \geq \delta(f_i, g_j) - \delta(f_i, h_k) \quad (8)$$

In words,  $\gamma(f_i, g_j, h_k)$  represents how long a particular job ran between the interval defined by the starting points of two other jobs: the execution time of the  $i$ th job on generator  $f$  between the start of the  $j$ th job on generator  $g$  and the start of the  $k$ th job on generator  $h$ . Given this, we can formulate the sufficient energy constraint as follows. Given watts  $W(g_j)$  produced by job  $j$  on generator  $g$  (where negative watts on a specified generator stand in for consumption) and  $f, g, h \in G; i \in 1 \dots N_f; j \in 1 \dots N_g; k \in 1 \dots N_h$ ,

$$\sum_{i=1}^{N_f} \gamma(f_i, g_j, h_k) W(f_i) \geq 0 \quad (9)$$

Give a cost function (in dollars, say) for the  $i$ th job on generator  $f$  as  $C(f_i)$ , we minimize the objective function

$$\sum_{i=1}^{N_f} \frac{\gamma(f_i, g_j, h_k) C(f_i)}{j(N_h - k)}. \quad (10)$$

Note that execution occurring between particular  $g_j$  and  $h_k$  will be counted multiple times; the division by  $j(N_h - k)$  normalizes the result.

## 4 The Big Problem

In my previous work on DVFS scheduling I used a similar formulation to approximate the maximum. A given job with backward dependences on other jobs could not be scheduled to start until all its dependences had completed. I represented the start time of this job as greater than or equal to the start time plus run time of each of its dependences. As the problem bounded the total time in the system and the objective function used scheduled time as an input, the calculated start time did not exceed the minimum solution.

UCP differs in one important respect: the max approximation must be applied at two levels,  $\delta$  and  $\gamma$ , and the objective function only operates on the latter. This means that if additional constraints are not able to force  $\delta$  to be very close to max, the solutions for  $\gamma$  may be feasible but useless.

The following place upper bounds on  $\delta$ :

$$\delta(g_j, h_k) \leq g_j \quad (11)$$

$$\delta(g_j, h_k) \leq T - h_k \quad (12)$$

$$\delta(g_j, h_k) \leq \delta(g_j, h_{k+1}) \quad (13)$$

$$\delta(g_{j+1}, h_k) \leq \delta(g_j, h_k). \quad (14)$$

If  $\delta(g_j, h_k)$  is positive, then  $\delta(h_k, g_j)$  should be constrained to be zero. If  $\delta(g_j, h_k)$  is zero, then  $\delta(h_k, g_j)$  may be zero (if  $g_j = h_k$ ) or positive.

Additionally, we can “sum the differences” as follows:

$$\sum_{i=1}^{N_f-1} \delta(f_{i+1}, g_j) - \delta(f_i, g_i) = T - g_i \quad (15)$$

$$\sum_{j=1}^{N_g-1} \delta(f_i, g_j) - \delta(f_i, g_{j+1}) = f_i. \quad (16)$$

To the best of my knowledge, these constraints on  $\delta$  are not sufficient to give useful answers to the UCP, nor are there a set of constraints that exist that would allow useful answers.

## References

- [1] J. M. Arroyo and A. J. Conejo. A parallel repair genetic algorithm to solve the unit commitment problem. *IEEE Transactions on Power Systems*, 17(4):1216–1224, Nov 2002.
- [2] A. Borghetti, A. Frangioni, F. Lacalandra, and C. A. Nucci. Lagrangian heuristics based on disaggregated bundle methods for hydrothermal unit commitment. *IEEE Transactions on Power Systems*, 18(1):313–323, Feb 2003.
- [3] Chuan-Ping Cheng, Chih-Wen Liu, and Chun-Chang Lie. Unit commitment by lagrangian relaxation and genetic algorithms. *IEEE Transactions on Power Systems*, 15(2):707–714, May 2000.
- [4] I. G. Damousis, A. G. Bakirtzis, and P. S. Dokopoulos. A solution to the unit-commitment problem using integer-coded genetic algorithm. *IEEE Transactions on Power Apparatus and Systems*, 19(2):1165–1172, May 2004.
- [5] D. Dasgupta and D. R. McGregor. Thermal unit commitment using genetic algorithms. *IEE Proceedings: Generation, Transmission and Distribution*, 141(5):459–465, Sep 1994.



- [6] Zwe-Lee Gaing. Discrete particle swarm optimization algorithm for unit commitment. *IEEE Power Engineering Society General Meeting*, Jul 2003.
- [7] X. Guan, P. B. Luh, and H. Yan. An optimization-based method for unit commitment. *International Journal of Electrical Power and Energy Systems*, 14(1):9–17, Feb 1992.
- [8] W. J. Hobbs, G. Hermon, S. Warner, and G. B. Shelbe. An enhanced dynamic programming approach for unit commitment. *IEEE Transactions on Power Apparatus and Systems*, 3(3):1201–1205, Aug 1988.
- [9] Shyh-Jier Huang and Ching-Lien Huang. Application of genetic-based neural networks to thermal unit commitment. *IEEE Transactions on Power Systems*, 12(2):654–660, May 1997.
- [10] K. A. Juste, H. Kita, E. Tanaka, and J. Hasegawa. An evolutionary programming solution to the unit commitment problem. *IEEE Transactions on Power Systems*, 14(4):1452–1459, Nov 1999.
- [11] Chao-An Li, R. B. Johnson, and A. J. Svoboda. A new unit commitment method. *IEEE Transactions on Power Systems*, 12(1):113–119, Feb 1997.
- [12] Tao Li and M. Shahidehpour. Price-based unit commitment: a case of lagrangian relaxation versus mixed integer programming. *IEEE Transactions on Power Systems*, 20(4):2015–2025, Nov 2005.
- [13] P. G. Lowery. Generating unit commitment by dynamic programming. *IEEE Transactions on Power Apparatus and Systems*, PAS-85(5):422–426, May 1966.
- [14] T. T. Maifeld and G. B. Sheble. Genetic-based unit commitment algorithm. *IEEE Transactions on Power Systems*, 11(3):1359–1370, Aug 1996.
- [15] A. H. Mantawy, Y. L. Abdel-Magid, and S. Z. Selim. A simulated annealing algorithm for unit commitment. *IEEE Transactions on Power Systems*, 13(1):197–204, Feb 1998.
- [16] A. H. Mantawy, Y. L. Abdel-Magid, and S. Z. Selim. Unit commitment by tabu search. *IEE Proceedings: Generation, Transmission and Distribution*, 145(1):56–64, Jan 1998.
- [17] A. H. Mantawy, Y. L. Abdel-Magid, and S. Z. Selim. Integrating genetic algorithms, tabu search, and simulated annealing for the unit commitment problem. *IEEE Transactions on Power Systems*, 14(3):829–836, Aug 1999.
- [18] A. Merlin and P. Sandrin. A new method for unit commitment at electricite de france. *IEEE Transactions on Power Apparatus and Systems*, PAS-102(5):1218–1225, May 1983.

- [19] K. Methaprayoon, C Yingvivatanapong, Wei-Jen Lee, and J. R. Liao. An integration of ann wind power estimation into unit commitment considering the forecasting uncertainty. *IEEE Transactions on Industry Applications*, 43(6):1441–1448, Nov 2007.
- [20] S. Mokhtari, J. Singh, and B. Wollenberg. A unit commitment expert system. *IEEE Transactions on Power Systems*, 3(1):272–277, Feb 1987.
- [21] W. Ongsakul and N. Petcharakas. Unit commitment by enhanced adaptive lagrangian relaxation. *IEEE Transactions on Power Systems*, 19(1):620–628, Feb 2004.
- [22] Z. Ouyang and S. M. Shahidehpour. An intelligent dynamic programming for unit commitment application. *IEEE Transactions on Power Apparatus and Systems*, 6(3):1203–1209, Aug 1991.
- [23] Z. Ouyang and S. M. Shahidehpour. A hybrid artificial neural network-dynamic programming approach to unit commitment. *IEEE Transactions on Power Systems*, 7(1):236–242, Feb 1992.
- [24] N. P. Padhy. Unit commitment using hybrid models: a comparative study for dynamic programming, expert system, fuzzy system and genetic algorithms. *International Journal of Electrical Power and Energy Systems*, 23(8):827–836, Nov 2001.
- [25] N. P. Padhy. Unit commitment: a bibliographical survey. *IEEE Transactions on Power Systems*, 19(2):1196–1205, May 2004.
- [26] C. K. Pang and H. C. Chen. Optimal short-term thermal unit commitment. *IEEE Transactions on Power Apparatus and Systems*, 95(4):1336–1346, Jul 1976.
- [27] W. L. Peterson and S. R. Brammer. A capacity based lagrangian relaxation unit commitment with ramp rate constraints. *IEEE Transactions on Power Systems*, 10(2):1077–1084, May 1995.
- [28] V. Petridis, S. Kazarlis, and A. Bakirtzis. Varying fitness functions in genetic algorithm constrained optimization: the cutting stock and unit commitment problems. *IEEE Systems, Man and Cybernetics, Part B: Cybernetics*, 28(5):629–640, Oct 1998.
- [29] G. K. Purushothama and L. Jenkins. Simulated annealing with local search: a hybrid algorithm for unit commitment. *IEEE Transactions on Power Systems*, 18(1):273–278, Feb 2003.
- [30] C. C. A. Rajan and M. R. Mohan. An evolutionary programming-based tabu search method for solving the unit commitment problem. *IEEE Transactions on Power Systems*, 19(1):577–585, Feb 2004.

- [31] C. W. Richter and G. B. Sheble. A profit-based unit commitment ga for the competitive environment. *IEEE Transactions on Power Systems*, 15(2):715–721, May 2000.
- [32] Barry Rountree, David K. Lowenthal, Shelby H. Funk, Vincent W. Freeh, Bronis R. de Supinski, and Martin Schulz. Bounding Energy Consumption in Large-Scale MPI Programs. In *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, Reno, Nevada, November 2007.
- [33] A. Rudolf and R. Bayriethner. A genetic algorithm for solving the unit commitment problem of a hydro-thermal power system. *IEEE Transactions on Power Systems*, 14(4):1460–1468, Nov 1999.
- [34] S. Ruzic and N. Rjakovic. A new approach for solving extended unit commitment problem. *IEEE Transactions on Power Systems*, 6(1):269–277, Feb 1991.
- [35] S. Saneifard, N. R. Prasad, and H. A. Smolleck. A fuzzy logic approach to unit commitment. *IEEE Transactions on Power Systems*, 12(2):988–995, May 1997.
- [36] H. Sasaki, M. Watanabe, J. Kubokawa, N. Yorino, and R. Yokoyama. A solution method of unti commitment by artificial neural networks. *IEEE Transactions on Power Systems*, 7(3):974–981, Aug 1992.
- [37] Subir Sen and D. P. Kothari. Optimal thermal generating unit commitment: a review. *International Journal of Electrical Power and Energy Systems*, 20(7):443–451, Oct 1998.
- [38] T. Senjyu, H. Yamashiro, K. Uezato, and T. Funabashi. A unit commitment problem by using genetic algorithm based on unit characteric classification. *IEEE Poer Engineering Society Winter Meeting*, pages 58–63, 2002.
- [39] G. B. Sheble. Solution of the unit commitment problem by the method of unit periods. *IEEE Transactions on Power Systems*, 5(1):257–260, Feb 1990.
- [40] Gerald B. Sheble and Timothy T. Maifeld. Unit commitment by genetic algorithm and expert system. *Electric Power Systems Research*, 30(2):115–121, Jul 1994.
- [41] D. N. Simopoulos, S. D. Kavatza, and C. D. Vournas. Unit commitment by an enhanced simulated annealing algorithm. *IEEE Transactions on Power Systems*, 21(1):68–76, Feb 2006.
- [42] Walter L. Snyder, H. David Powell, and John C. Rayburn. Dynamic programming approach to unit commitment. *IEEE Transactions on Power Systems*, 2(2):339–348, May 1987.

- [43] Chung-Ching Su and Yuan-Yih Hsu. Fuzzy dynamic programming: an application to unit commitment. *IEEE Transactions on Power Systems*, 6(3):1231–1237, Aug 1991.
- [44] K. S. Swarup and S. Yamashiro. Unit commitment solution methodology using genetic algorithm. *IEEE Transactions on Power Systems*, 17(1):87–91, Feb 2002.
- [45] T. O. Ting, M. V. C. Rao, and C. K. Loo. A novel approach for unit commitment problem via an effective hybrid particle swarm optimization. *IEEE Transactions on Power Apparatus and Systems*, 21(1):411–418, Feb 2006.
- [46] Jorge Valenzuela and Alice E. Smith. A seeded memetic algorithm for large unit commitment problems. *Journal of Heuristics*, 8(2):173–195, May 2000.
- [47] S. Virmani, E. C. Adrian, K. Imhof, and S. Mukherjee. Implementation of a lagrangian relaxation based unit commitment problem. *IEEE Transactions on Power Apparatus and Systems*, 4(4):1373–1380, Nov 1989.
- [48] M. P. Walsh and M.J. O’Malley. Augmented hopfield network for unit commitment and economic dispatch. *IEEE Transactions on Power Systems*, 12(4):1765–1774, Nov 1997.
- [49] Hong-Tzer Yang, Pai-Chuan Yang, and Ching-Lien Huang. A parallel genetic algorithm approach to solving the unit commitment problem: implementation on the transputer networks. *IEEE Transactions on Power Systems*, 12(2):661–668, May 1997.
- [50] Qiaozhu Zhai, Xiaohong Guan, and Jian Cui. Unit commitment with identical units successive subproblem solving method based on lagrangian relaxation. *IEEE Transactions on Power Systems*, 17(4):1250–1257, Nov 2002.
- [51] F. Zhuang and F. D. Galiana. Towards a more rigorous and practical unit commitment by lagrangian relaxation. *IEEE Transactions on Power Systems*, 3(2):763–773, May 1988.
- [52] F. Zhuang and F. D. Galiana. Unit commitment by simulated annealing. *IEEE Transactions on Power Systems*, 5(1):311–318, Feb 1990.